

Network cross-validation by edge sampling

Tianxi Li, Elizaveta Levina, Ji Zhu
Department of Statistics, University of Michigan

January 3, 2017

Abstract

Many models and methods are now available for network analysis, but model selection and tuning remain challenging. Cross-validation is a useful general tool for these tasks in many settings, but is not directly applicable to networks since splitting network nodes into groups requires deleting edges and destroys some of the network structure. Here we propose a new network cross-validation strategy based on splitting edges rather than nodes, which avoids losing information and is applicable to a wide range of network problems. We provide a theoretical justification for our method in a general setting, and in particular show that the method has good asymptotic properties under the stochastic block model. Numerical results on both simulated and real networks show that our approach performs well for a number of model selection and parameter tuning tasks.

1 Introduction

Methods for statistical analysis of network data have received a lot of attention because of the wide-ranging applications of network analysis. There is now a large body of work on methods and models for networks, including the stochastic block model (SBM) [21], the degree-corrected stochastic block model (DCSBM) [25], and the latent space model [20] to name a few. While this gives the practitioner a wide range of model choices, there is a lot less work on the crucial problem of how to select the best model for the data, as well as how to choose tuning parameters for the selected model which is often necessary. In some specific problems, progress has been made recently – for instance, in the much-studied problem of community detection. Community detection is the problem of clustering network nodes into groups, and most of the methods proposed over the last twenty years or so require the number of communities K as input. In practice, however, K is often unknown and has to be estimated, and substantial progress has been made on this problem quite recently [28, 35, 5, 32, 39, 44, 11, 29], mostly under the SBM assumption. Above choosing K , there is more general model selection problem in the community detection problem and beyond, which may require comparing two models that are not necessarily nested. [46] developed a likelihood ratio type test to select between SBM and DCSBM, although only under the assumption that both models use the same K . The cross-validation approach of [11] is more general and can compare SBM and DCSBM and choose K simultaneously. Beyond choosing the number of communities and deciding on the degree correction, many network analysis methods rely on additional tuning parameters, for example, regularized spectral clustering

(RSC) [10, 2, 37] and covariate-assisted spectral clustering (CASC) [6]. While some tuning methods for these specific methods have been proposed (for example, in [24] for RSC and [6] for CASC), a general strategy for selecting tuning parameters in network settings is lacking.

In classical settings where the data points are assumed to be an i.i.d. sample, cross-validation is one of the most general and appealing ways for model selection and tuning. In general, cross-validation works by splitting the training data into multiple parts (folds), holding out one fold at a time as a test set on which to compute the error of the model fitted on the remaining folds, averaging the errors across all folds to obtain the cross-validation error, and choosing the model or the tuning parameter value that minimizes this error. Cross-validation relies on two key conditions: i) data splitting results in independent datasets, so that cross-validation mimics the true generalization error; and ii) after holding out one fold, the remaining data points are sufficient to fit the same model. For network data, splitting nodes may result in violating both of these conditions: there is generally dependence between nodes induced by their shared edges, and more importantly, splitting nodes leads to deleting edges, and thus may alter the fit of the model in significant ways. A valid strategy for network splitting and validation under Bayesian framework was considered by [19], but this is computationally prohibitive even for networks of moderate sizes. One novel way for network cross-validation (NCV) was recently proposed by [11], who create cross-validation folds by block-wise splitting. NCV is effective for community detection tasks under either the SBM or the DCSBM, and is able to compare these two models. However, NCV relies on the presence of blocks, which limits its applicability for general networks.

In this paper, we propose a new *edge* cross-validation (ECV) framework for networks. Instead of splitting nodes, we split pairs of nodes into different folds. Treating the network after removing the information on some node pairs as an incompletely observed network, we use low rank matrix completion to estimate relevant model parameters. This reconstructed network has the same rate of concentration around its expectation as the full network adjacency matrix, thus allowing for efficient model fitting after edge-splitting. Our method is valid under a very general class of network models including both directed and undirected networks. As special applications, we also propose model-free methods for selecting the number of communities K and the tuning parameter for regularized spectral clustering.

The rest of the paper is organized as follows. Section 2 introduces the new edge-based cross-validation algorithm ECV and compares it to node-based NCV [11] for the case of block models. It also introduces the special cases of choosing the number of communities and tuning spectral clustering. Section 3 presents a general error bound of ECV estimation. Section 4 presents extensive simulation studies of ECV and its competitors for the tasks of model selection in block models and tuning regularized spectral clustering. Section 5 presents two real-world applications and Section 6 concludes with discussion.

2 The edge cross-validation (ECV) algorithm

2.1 Notation and model

Let $\mathcal{V} = \{1, 2, \dots, n\} =: [n]$ denote the node set of a network, and let A be its $n \times n$ adjacency matrix, where $A_{ij} = 1$ if there is an edge from node i to node j and 0 otherwise. For undirected networks, A is a symmetric matrix. Let $D = \text{diag}(d_1, d_2, \dots, d_n)$ be the diagonal matrix with node degrees $d_i = \sum_j A_{ij}$ on the diagonal. The (normalized) Laplacian

of a network is defined to be $L = D^{-1/2}AD^{-1/2}$. Finally, we write I_n for the $n \times n$ identity matrix and $\mathbf{1}_n$ for $n \times 1$ column vector of ones, suppressing the dependence on n when it is clear from the context. For any matrix M , we use $\|M\|$ to denote its spectral norm and $\|M\|_F$ to denote its Frobenius norm.

We view A as a single random realization of independent Bernoulli variables, with $\mathbb{E}A = M$, where M is a matrix of probabilities. For undirected networks, we further assume M is symmetric and the unique edges A_{ij} are independent Bernoulli variables, with $A_{ji} = A_{ij}$. The key assumption underlying our method is that M is a low-rank matrix (or close to one). This class includes a wide range of network models, including stochastic block models and degree-corrected stochastic block models discussed in more detail in Section 2.4.

2.2 The ECV procedure

For notational simplicity, we present the algorithm for directed networks; the only modification needed for undirected networks is treating node pairs (i, j) and (j, i) as one pair.

The key insight of ECV is to split node pairs rather than nodes. We randomly sample node pairs (regardless of the actual value of A_{ij}) with a fixed probability to be in the holdout test set. Note that by assumption the values of A corresponding to the test pairs are independent of those corresponding to the training pairs. Next, we need to fit the model of interest, whatever it may be, using only the training pairs of nodes. The matrix we have available for training, with a certain proportion of entries missing completely at random, is the classic setting for matrix completion. Let $\Omega \subset [n] \times [n]$ be the index set of the remaining training entries and correspondingly, Ω^\perp be the index set of the hold-out entries. Define the operator $P_\Omega : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ by

$$(P_\Omega A)_{ij} = A_{ij}\mathbf{I}\{(i, j) \in \Omega\}$$

We then “reconstruct” the full network matrix by finding a low-rank approximation via solving the following optimization problem:

$$\begin{aligned} \hat{M} &= \arg \min_M \|P_\Omega A - P_\Omega M\|_F^2 \\ \text{s.t. rank}(M) &= K \\ 0 &\leq M_{ij} \leq \min\{1, \frac{d}{n}\}. \end{aligned} \tag{1}$$

The bound d/n is optional and can be used to incorporate any further assumptions on the values of M ; in the absence of such assumptions, one can set $d = n$ and the constraint simply indicates that M_{ij} ’s are probabilities. Without the second constraint, this problem is a generalized SVD problem for incomplete matrices and is in a similar spirit of the matrix completion problem discussed by [9]. Evidently, this approach implicitly relies on the assumption that A or at least its expectation can be reasonably well approximated by a low rank matrix, which in practice has been shown in a wide range of applications. Theoretical guarantees relying on this assumption will be made explicit in Section 3. Further, if the value of K is unknown, as is usually the case in practice, it can be chosen as part of the same the cross-validation procedure.

In practice, (1) is often solved by convex relaxations or approximations [7]. We use a constrained fixed point iteration method that is similar to the *hardImpute* algorithm of [34]

(see Appendix B), with the second constraint incorporated in the spirit of [27].

Generally speaking, any reasonable matrix completion method should work for the purposes of ECV. For instance, the *OptSpace* method proposed by [26] is an alternative way to complete the matrix which can be shown to have the same theoretical guarantees. Since most matrix completion methods are based on formulations equivalent to (1) and their corresponding convex relaxations, in this paper, we take (1) as our matrix completion criterion. One could argue that given the nature of A it might be even more appropriate to consider binary rather than general matrix completion methods, also known as 1-bit matrix completion [15, 8, 3]. However, 1-bit matrix completion methods are less appealing in practice as they are generally much more computationally demanding than the Frobenius norm-based completion which has been used for very large matrices [34].

Finally, once we obtain the completed matrix \hat{M} , we can fit the candidate models on the training data and evaluate loss on the hold-out entries of A , just as in standard cross-validation. There may be more than one way to evaluate the loss on \hat{M} if the loss function itself is designed for binary input; we will elaborate on the details in specific examples. The general algorithm is summarized as Algorithm 1 below.

Algorithm 1 (The general ECV procedure). Input: an adjacency matrix A , a loss function L , a set \mathcal{C} of Q candidate models or parameter values to select from, the training proportion p , and the number of replications N .

1. For $m = 1, \dots, N$
 - (a) Randomly choose a subset of node pairs $\Omega \subset \mathcal{V} \times \mathcal{V}$, selecting each pair independently with probability p .
 - (b) Apply matrix completion to the pair (A, Ω) to obtain \hat{M} .
 - (c) For each of the candidate models $q = 1, \dots, Q$, fit the model on the completed matrix \hat{M} , and evaluate its loss $L_q^{(m)}$ by applying the loss function L with the estimated parameters to $A_{ij}, (i, j) \in \Omega^\perp$.
2. Let $L_q = \frac{1}{N} \sum_{m=1}^N L_q^{(m)}$. Return $\hat{q} = \operatorname{argmin}_q L_q$ (the best model out of the candidate set \mathcal{C}).

2.3 Model-free rank estimators

The rank constraint for the matrix completion problem is typically unknown, and in practice we need to choose or estimate it in order to apply ECV. Selection of K can itself be treated as a model selection problem, considering that the completed matrix \hat{M} itself is a low rank approximation to the underlying probability matrix M without any further model assumptions. We can thus compare \hat{M} to A in some way in order to select the best rank.

One natural approach is to directly compare the values of \hat{M} on the held-out entries of A . For instance, we can use the sum of squared imputation errors,

$$\text{SSE} = \sum_{(i,j) \in \Omega^\perp} (A_{ij} - \hat{A}_{ij})^2,$$

or alternatively compute the binomial deviance on this set, and pick the value of K that minimizes this.

Another approach is to consider how well \hat{M} performs on predicting links. We can predict $\hat{A}_{ij} = \mathbf{I}\{\hat{M}_{ij} > c\}$ for all entries in the hold-out set Ω^\perp for a threshold c , and vary c from 0 to 1 to obtain a sequence of link prediction results. A common measure of prediction performance is the area under the ROC curve (AUC), which compares false positive rates to true positive rates for all values of c , with perfect prediction corresponding to AUC of 1, and random guessing to 0.5. We then select the rank K which maximizes the AUC.

In practice, we have observed that both the imputation error measure and the AUC work well in general rank estimation tasks. They perform comparably to likelihood-based methods for block models most of the time, though the AUC has some over-selection problem in very easy settings due to the nature of this metric (see Section 4 for details).

2.4 Model selection for block models

The stochastic block model (SBM) is perhaps the most widely used undirected network model with communities that assumes community structure in the probability matrix M . In particular, it assumes that $M = ZBZ^T$ where $B \in [0, 1]^{K \times K}$ is a symmetric probability matrix and $Z \in \{0, 1\}^{n \times K}$ has exactly one “1” in each row, with $Z_{ik} = 1$ if node i belongs to community k . We further use $\mathbf{c} = \{c_1, \dots, c_n\}$ to denote the vector of membership for all nodes, such that $c_i \in [K]$. Then the probability of having an edge between nodes i and j is $P(A_{ij} = 1) = B_{c_i c_j}$.

One of the commonly pointed out limitations of the SBM is that it forces equal expected degrees for all the nodes in the same community, therefore ruling out “hubs”. The degree corrected stochastic block model (DCSBM) corrects this shortcoming of the SBM by allowing nodes to have individual “degree parameters”, θ_i associated with each node i and let $\Theta = \text{diag}(\theta_1, \dots, \theta_n)$. The DCSBM then assumes $\mathbb{E}A = M = \Theta ZBZ^T \Theta$ (a constraint is needed on Θ to ensure identifiability, with different authors choosing different versions; here we follow [25] and assume $\sum_{c_i=k} \theta_i = 1$, for each $k \in [K]$). Equivalently, we can write $P(A_{ij} = 1) = \theta_i \theta_j B_{c_i c_j}$. Note that both the SBM and the DCSBM assume the probability matrix M has rank K . Throughout this paper, we refer to the SBM and the DCSBM together as block models.

The choice of fitting method is not crucial for model selection, and many methods are now available for fitting the SBM and DCSBM such as [25, 48, 4] and [2]; for simplicity, we use one of the simplest, fastest, and most common methods, spectral clustering on the Laplacian $L = D^{1/2}AD^{1/2}$, where D is the diagonal matrix of node degrees. Spectral clustering computes K leading eigenvectors of L , arranged in a $n \times K$ matrix U , and applies the K -means clustering algorithm to the rows of U to obtain cluster assignments for the n nodes.

Spectral clustering enjoys asymptotic consistency under the SBM when the average degree grows at least as fast as $\log n$ [38, 33, 40]. There are also variants of spectral clustering that are consistent under the DCSBM, such as the spherical spectral clustering [37, 33] that normalizes the row vectors of U before applying K -means and the SCORE method [23] that divides each column of U by the first column of U .

Note that since both SBM and DCSBM are undirected network models, we use the undirected variant of ECV, selecting edges at random from the set of pairs (i, j) with $i < j$ only and including the pair (j, i) whenever (i, j) is selected. Once node memberships are estimated, the other parameters are easy to estimate by conditioning on node labels. Specif-

ically, for the SBM we simply take the MLE conditional on the node labels evaluated on the available node pairs. Let $\hat{C}_k = \{i : (i, j) \in \Omega, \hat{c}_i = k\}$ be the estimated member sets for each group $k = 1, \dots, K$. Then we can estimate the entries of the probability matrix B as

$$\hat{B}_{kl} = \frac{\sum_{(i,j) \in \Omega} A_{ij} 1(\hat{c}_i = k, \hat{c}_j = l)}{\hat{n}_{kl}^\Omega} \quad (2)$$

where

$$\hat{n}_{kl}^\Omega = \begin{cases} |(i, j) \in \Omega : \hat{c}_i = k, \hat{c}_j = l| & \text{if } k \neq l \\ |(i, j) \in \Omega : i < j, \hat{c}_i = \hat{c}_j = k| & \text{if } k = l. \end{cases}$$

Under DCSBM, the probability matrix can be estimated similarly to [25, 48] and [24] via the Poisson approximation, letting

$$\hat{O}_{kl}^* = \sum_{(i,j) \in \Omega} A_{ij} 1(\hat{c}_i = k, \hat{c}_j = l) \quad (3)$$

and

$$\hat{\theta}_i = \frac{\sum_{j: (i,j) \in \Omega} A_{ij}}{\sum_{k=1}^K \hat{O}_{\hat{c}_i, k}^*}.$$

The final probability matrix is then estimated by

$$\hat{P}_{ij} = \hat{\theta}_i \hat{\theta}_j \hat{O}_{\hat{c}_i \hat{c}_j}^* / p. \quad (4)$$

Note that the final probability estimate is scaled by p to reflect the random missing edge mechanism, which makes it slightly different from the estimator for the fully observed DCSBM as in [25]. Note that this rescaling happens automatically in the SBM estimate (2) since the sums in both the numerator and the denominator range over Ω only.

Finally, we need to specify loss functions to be evaluated on the hold-out set. The natural loss functions for these models are either the squared error loss

$$L_2(A, \hat{M}) = \sum_{i < j, (i,j) \in \Omega^\perp} (A_{ij} - \hat{M}_{ij})^2,$$

or, to match the maximum likelihood based parameter estimates, the binomial deviance function

$$L_d(A, \hat{M}) = - \sum_{i < j, (i,j) \in \Omega^\perp} \left[A_{ij} \log(\hat{M}_{ij}) - (1 - A_{ij}) \log(1 - \hat{M}_{ij}) \right].$$

In practice, we observed that binomial deviance works slightly better in model selection under both SBM and DCSBM.

The model selection question for block models includes the choice of SBM vs DCSBM together with the choice of K for each model. Suppose we want to select either SBM or DCSBM with the number of communities in both models ranging from 1 to K_{\max} . The candidate set of models in Algorithm 1 is then $\mathcal{C} = \{\text{SBM-}K, \text{DCSBM-}K, K = 1, \dots, K_{\max}\}$ where the number after the model name is the number of communities. The ECV algorithm for block model selection is summarized below as Algorithm 2.

Algorithm 2. Input: an adjacency matrix A , candidate model set \mathcal{C} , the training proportion

p , and the number of replications N

1. For $m = 1, \dots, N$
 - (a) Randomly choose a subset of node pairs Ω : selecting each pair (i, j) , $i < j$ independently with probability p , and adding the pair (j, i) if (i, j) is selected.
 - (b) For $K = 1, \dots, K_{\max}$,
 - i. Apply matrix completion to the pair (A, Ω) with rank constraint K to obtain \hat{M}_K .
 - ii. Run spectral clustering to obtain the estimated SBM membership vector $\hat{\mathbf{c}}_{1,K}^{(m)}$, and spherical spectral clustering to obtain the estimated DCSBM $\hat{\mathbf{c}}_{2,K}^{(m)}$.
 - iii. Estimate the probability matrix $\hat{M}_{q,K}^{(m)}$ for $q = 1, 2$ based on $\hat{\mathbf{c}}_{q,K}^{(m)}$ and evaluate its loss $L_{q,K}^{(m)}$ by applying the loss function L with the estimated parameters to A_{ij} , $(i, j) \in \Omega^\perp$.
2. Let $L_{q,K} = \frac{1}{N} \sum_{m=1}^N L_{q,K}^{(m)}$. Return $(\hat{q}, \hat{K}) = \arg \min_{q=1,2} \min_{K=1,\dots,K_{\max}} L_{q,K}$ as the best model (with $\hat{q} = 1$ indicating SBM and $\hat{q} = 2$ indicating DCSBM).

The network cross-validation (NCV) algorithm by [11] was introduced explicitly for the purpose of model selection in block models, and thus it is of interest to consider the differences with ours. The NCV algorithm splits node pairs by first randomly splitting nodes; once the nodes are divided at random into two groups \mathcal{N}_1 and \mathcal{N}_2 , and the pairs (i, j) corresponding to $i \in \mathcal{N}_1$ and $j \in \mathcal{N}_1 \cup \mathcal{N}_2$ are arranged into a rectangular matrix. The right singular vectors of this matrix are then passed on to either spectral clustering for SBM or spherical spectral clustering for DCSBM to estimate node labels. The SBM model parameters can be estimated by standard estimators. However, standard estimators of DCSBM model parameters cannot be easily extended to a rectangular matrix, so a special estimator is proposed in [11]. The node pairs (i, j) corresponding to $i, j \in \mathcal{N}_2$ are then used as a test set to evaluate the loss function and choose the best model.

The ECV is more general than the NCV, since in general it works with any low-rank approximation and does not rely on block structure in the data, and also works for both directed and undirected networks, where NCV is for undirected networks only. As NCV does not recover the adjacency matrix, it cannot be used to evaluate methods that are based on certain transformations of the adjacency matrix, such as the problem in Section 2.5. Further, ECV is less likely to create isolated nodes in the training sample, which are useless in estimation. To see this, consider the following simple calculation: assume that a given node i has degree d , and that all its d neighbors also have degree d . Suppose we apply N -fold NCV by deleting n/N rows of A , and hold out a matching number of entries at random via ECV. Let p_n and p_e be the probabilities that all neighbors of the given node i are assigned to the hold-out entries by NCV and ECV, respectively. Then a simple combinatorial calculation combined with Stirling's formula shows that for large n , the ratio of the two probabilities is approximately

$$p_e/p_n \approx e^{d/N^2} / N^d.$$

This ratio achieves its maximum 0.64 when $N = 2$ and $d = 1$ and can be much smaller if $N > 2, d > 1$. Table 1 shows p_e/p_n when $n = 300$ and $N = 3$, for different d .

Although this example is a simplified calculation for one fixed node, it shows an important advantage of ECV over NCV under the block models, since isolated nodes are assigned to

Table 1: Ratio between p_e and p_n for $n = 300$, $N = 3$, and different d , where p_e and p_b are the probabilities that a node with d neighbors becomes isolated in the training set in ECV and NCV, respectively.

d	1	2	3	4	5
p_e/p_n	0.339	0.113	0.035	0.012	0.004

blocks randomly and decrease overall accuracy. In simulations, we also observed that ECV is much less likely to result in isolated nodes than NCV.

2.5 Parameter tuning in regularized spectral clustering

Regularized spectral clustering has been proposed to improve performance of spectral clustering in sparse networks, but regularization itself frequently depends on a tuning parameter that has to be selected correctly in order to achieve the improvement. Several different regularizations have been proposed and analyzed [10, 2]. ECV can be used to tune all of them, but for concreteness here we focus on the proposal by [2], who replace the usual normalized graph Laplacian $L = D^{-1/2}AD^{-1/2}$, where D is the diagonal matrix of node degrees, by the Laplacian computed from regularized adjacency matrix

$$A_\tau = A + \tau \cdot \hat{d}/n \mathbf{1}\mathbf{1}^T \quad (5)$$

where \hat{d} is the average node degree and τ is a tuning parameter, typically within $[0, 1]$. The scale of the multiplier is motivated by theoretical results under the SBM [17, 31], and the method is considered not sensitive to the choice of τ within a fairly wide range. Nevertheless, in practice we have observed that different τ 's give very different clustering results, for instance in the example in Section 5.2. [24] proposed a data-driven way to select τ called DKest based on theoretical bounds obtained under SBM and DCSBM. Using ECV allows us to propose an alternative general way of selecting τ which does not rely on the block model assumption, and as we will show in later sections, remains valid even when the block models are clearly wrong and DKest fails.

Choosing a good τ is expected to give good clustering accuracy, defined as proportion of correctly clustered nodes under the best cluster match,

$$\max_{\hat{\mathbf{c}}^p \in \text{perm}(\hat{\mathbf{c}})} |\{i \in [n], \hat{c}_i^p = c_i\}|/n.$$

We can directly use Algorithm 1 with the candidate set \mathcal{C} being a grid of τ values as long as we can specify a loss function. Since we would like to have a general method not relying on the block model assumption, we need a model-free loss function which can be applicable even when the block model does not hold (as in the example in Section 5.2). In general, finding a good loss function for cross-validation in clustering is difficult, and while there is some work in the usual clustering setting [43, 41, 42], it has not been discussed much in the network setting, and the loss function we propose next, one of a number of reasonable options, may be of independent interest.

For any cluster label vector \mathbf{c} , the set of node pairs $\mathcal{V} \times \mathcal{V}$ will be divided into $K(K+1)/2$ classes defined by $H(i, j) = (c_i, c_j)$. Notice that we treat each $H(i, j)$ as an unordered pair, since the network is undirected in spectral clustering. To compare two vectors of labels \mathbf{c}_1 and \mathbf{c}_2 , we can compare their corresponding partitions H_1 and H_2 by computing co-

clustering difference (CCD) or normalized mutual information (NMI) between them [?]. For instance, the co-clustering matrix for H_1 is defined to be the $n^2 \times n^2$ matrix G_1 such that $G_{1,(j-1)n+i,(q-1)n+p} = \mathbf{I}\{H_1(i, j) = H_1(p, q)\}$. It gives whether or not two edges are in the same partition of H_1 . Then the CCD between H_1 and H_2 is defined as the squared Frobenius norm of the difference between the two co-clustering matrices

$$\text{CCD}(H_1, H_2) = \|G_1 - G_2\|_F^2/2.$$

We apply this measure to choose the tuning parameter τ as follows: for each split $m = 1, 2, \dots, N$ of ECV and each candidate value of τ , we complete the adjacency matrix after removing the hold-out entries in the set Ω_m^\perp and estimate cluster labels $\hat{c}_\tau^{(m)}$ and the corresponding partition $\hat{H}_\tau^{(m)}$ by regularized spectral clustering on the completed matrix with the candidate value of τ . We also compute \hat{H}_τ , the partition corresponding to regularized spectral clustering on the full adjacency matrix with the same value of τ . Then we choose τ by comparing these partitions constrained on the hold-out set,

$$\hat{\tau} = \arg \min_{\tau \in \mathcal{C}} \sum_{m=1}^N \text{CCD}(\hat{H}_{\tau, \Omega_m^\perp}^{(m)}, \hat{H}_{\tau, \Omega_m^\perp}).$$

Intuitively, if τ is a good value, the label vectors that generate $\hat{H}_{\tau, \Omega_m^\perp}^{(m)}$ and $\hat{H}_{\tau, \Omega_m^\perp}$ should both be close to the truth (Theorem 1), and so the co-clustering matrices should be similar; if τ is a bad choice, then both label vectors will contain more (non-matching) errors, and the corresponding CCD will be larger

3 Theoretical guarantees

For simplicity, we state the theoretical results for a single split of ECV. The results are obtained under the following assumptions on the matrix $M = \mathbb{E}A$:

Assumption A1. $\text{rank}(M) = K$.

Assumption A2. $\max_{ij} M_{ij} \leq d_0/n$ for some positive d_0 .

Note that A2 can be satisfied trivially by setting $d_0 = n$. However, in many network models the entries of M are assumed to be $o(1)$ in order to avoid a dense graph, and the bounds to be introduced in this section can be improved if additional information about d_0 is available.

Recall that the matrix \hat{M} is recovered from the sampled edges by solving problem (1) i.e.

$$\hat{M} = \arg \min_M \|P_\Omega A - P_\Omega M\|_F^2 \text{ s.t. } \text{rank}(M) = K, 0 \leq M_{ij} \leq \min\{1, d/n\},$$

and thus satisfies A1 and A2 by construction, with $d_0 = d$ and d is a tuning parameter. Theorem 1 gives a bound on the error in this recovered matrix.

Theorem 1 (Concentration of ECV completed adjacency matrices). Let M be a probability matrix satisfying A1 and A2, and A an adjacency matrix of a network with independent edges such that $\mathbb{E}(A) = M$. Let Ω be a set of entries of A selected independently with probability $p \geq C_1 \log n/n$ for some absolute constant C_1 . If $d \geq \max\{d_0, C_2 \log(n)\}$ for some absolute constant C_2 , then for any $\delta > 0$, with probability at least $1 - 2n^{-\delta}$, the

solution of (1) i.e. \hat{M} satisfies

$$\frac{\|\hat{M} - M\|_F^2}{n^2} \leq C \frac{K^4 d}{n^2 p^3}. \quad (6)$$

where $C = C(\delta, C_1, C_2)$ is a constant.

The proof of the theorem is given in Appendix A. Note that the error bound for 1-bit matrix completion from [3] is $O(\frac{K^3}{np^4})$. Fixing K and p , we have an additional factor of $O(\frac{d}{n})$ which is $o(1)$ except for unrealistic graphs, due to the fact that we use the better concentration bound for network adjacency matrices that are not extremely sparse. We have a weaker requirement as well as a better rate on p compared to [3], but pay for a higher rate on K . The optimal error rate for general matrices with bounded values is $O(\frac{K}{pn})$ as nicely summarized by [27]. Note that in the cross-validation setting, p is controlled and can be treated as a constant, which makes our bound lower than the general matrix bound as long as $K^3 d = o(n)$.

Since the matrix spectral norm is bounded above by its Frobenius norm, (6) implies the rate of concentration of \hat{M} around M is the same as that for the full adjacency matrix [33, 12, 30] if one assumes K and p is fixed. Thus with a properly chosen p we can think of \hat{M} being as close to the truth as the full adjacency matrix A . The theorem implies one should use the smallest possible d in solving (1) as long as $d \geq \max\{d_0, C_2 \log n\}$. For simplicity, all numerical results in this paper are based on setting $d = n$ which gives $0 \leq M_{ij} \leq 1$. In certain situations such as the setting of Corollaries 1 and 2, one may be able to pick the best d . Numerically we did not observe significant differences in performance, however.

The general Theorem 1 can be stated more explicitly under the SBM and DCSBM. In following discussions, we always treat p and K as constant numbers. For SBM, we make the following standard assumptions:

Assumption A3. The probability matrix $B^{(n)} = \rho_n B_0$, where B_0 is a $K \times K$ symmetric nonsingular matrix with all entries lying in $[0, 1]$. Therefore the expected node degree is of the order $\lambda_n = n\rho_n$.

Assumption A4. There exists a constant $\gamma > 0$ such that $\min_k n_k > \gamma n$ where $n_k = |\{i : c_i = k\}|$.

Note that under A3 and A4, all entries of M are of the same order. Thus we can use

$$\bar{d} = n \sum_{(i,j) \in \Omega} A_{ij} / |\Omega|$$

to obtain an order estimate of d_0 to ensure a better concentration bound in Theorem 1. This concentration is enough to ensure success of many methods, such as spectral clustering. Note that there are many different K -means algorithms that can be used in spectral clustering. Here we assume the K -means algorithm is the same as in [33].

Corollary 1 (ECV network recovery and spectral clustering under the SBM). Let A be an adjacency matrix generated from an SBM satisfying A3 and A4 with K blocks as K being fixed, and $M = \mathbb{E}A$. Let \hat{M} be the solution of (1) with $d = \kappa \bar{d}$ for a large enough constant κ . If the expected node degree $\lambda_n \geq C_1 \log(n)$, then

1. For any $\delta > 0$, there exists a constant $C = C(\delta, C_1)$ such that with probability at least $1 - 2n^{-\delta}$

$$\|\hat{M} - M\| \leq C\sqrt{\lambda_n}.$$

2. Let $\hat{\mathbf{c}}$ be the output of spectral clustering on \hat{M} . Then $\hat{\mathbf{c}}$ coincides with the true \mathbf{c} on all but $O(n\lambda_n^{-1})$ nodes (up to a permutation of block labels), with probability tending to one.

To obtain analogous results for the DCSBM, we need one more standard assumption on the degree parameters, similar to [23, 33, 11].

Assumption A5. $\min_i \theta_i \geq \theta_0$ for some constant $\theta_0 > 0$ and $\sum_{i:c_i=k} \theta_i = 1$ for all $k \in [K]$.

Corollary 2 (ECV network recovery and spectral clustering under the DCSBM). Let A be an adjacency matrix from a DCSBM satisfying A3, A4 and A5 with K blocks for a fixed K , and $M = \mathbb{E}A$. Let \hat{M} be the solution of (1) with $d = \kappa\bar{d}$ for a large enough constant κ . If the expected node degree $\lambda_n \geq C_1 \log(n)$, then

1. For any $\delta > 0$, there exists a constant $C = C(\delta, C_1)$ such that with probability at least $1 - 2n^{-\delta}$

$$\|\hat{M} - M\| \leq C\sqrt{\lambda_n}.$$

2. Let $\hat{\mathbf{c}}$ be the output of spherical spectral clustering on \hat{M} . Then $\hat{\mathbf{c}}$ coincides with the true \mathbf{c} on all but $O(n\lambda_n^{-1/2})$ nodes (up to a permutation of block labels), with probability tending to one.

4 Numerical performance evaluation

For simulated networks, we evaluate the performance of ECV against NCV and other relevant competitors in three applications: model selection for block models (SBM vs DCSBM and the choice of K), tuning regularized spectral clustering, and estimating rank for a general low-rank network model.

4.1 Model selection for block models

When the correct model (SBM or DCSBM) is known/assumed, there are multiple methods available for selecting K and can be included in comparisons along with general cross-validation methods. We compare the following cross-validation procedures: the ECV method we proposed with binomial deviance as the loss function (ECV-dev), the NCV of [11] with the same loss function (NCV-dev), and the model-free ECV with the SSE and the AUC as loss functions described in Section 2.3 (ECV-SSE and ECV-AUC respectively). Additionally, we include three methods derived specifically for choosing K under the block models, which we would expect to be at least as accurate as the cross-validation methods considering that they use the true model and cross-validation does not. The method of [44] is a BIC-type criterion (LR-BIC) based on an asymptotic analysis of the likelihood ratio statistic. Another BIC-type method proposed by [39] is based on the composite likelihood (CL-BIC) (implementation available on the authors' website). Both LR-BIC and CL-BIC require an estimate of cluster labels, which we obtain from spectral clustering. From the

class of methods proposed by [29], we include the variant based on the Bethe-Hessian matrix with moment correction (BHmc).

The setting for all the simulated networks in this section is as follows: there are $K = 3$ communities and $n = 300$ nodes. All results are based on 200 replications. For the DCSBM, node degree parameters θ_i , $i = 1, \dots, n$ are generated from the power law distribution with lower bound 1 and scaling parameter 5; for the SBM, $\theta_i \equiv 1$ for all i . Let $B_0 = (1-\beta)I + \beta\mathbf{1}\mathbf{1}^T$ and $B \propto \Theta B_0 \Theta$, so that β is the out-in ratio (the ratio of between-block probability and within-block probability), and the scaling is selected so that the average node degree is λ . Let $\pi = (\pi_1, \pi_2, \pi_3)$ be the proportions of nodes in the three communities. We then vary three aspects of the model:

1. Sparsity: vary the expected average degree λ of the network from 4 to 40, fixing $\pi = (1/3, 1/3, 1/3)$ and $\beta = 0.2$.
2. Community size: let $\pi = 1/3 \cdot (1-t, 1, 1+t)$ and vary t from 0 to 0.5, fixing $\lambda = 40$ and $\beta = 0.2$.
3. Out-in ratio: vary β from 0 to 0.7, fixing $\lambda = 40$ and $\pi = (1/3, 1/3, 1/3)$.

The performance is evaluated on three different model selection tasks set by [11]: choosing K when the true model is known, choosing between SBM and DCSBM without knowing K , and choosing the number of communities K conditioning on correctly choosing SBM or DCSBM.

Figure 1 shows the results for choosing K when the true model is known, under the SBM (top row) and the DCSBM (bottom row). On this task, we observe that the model-based ECV (ECV-dev) always outperforms or is similar to the model-based NCV (NCV-dev), and most of the time it is quite comparable to model-based methods, occasionally outperforming them (for DCSBM as the out-in ratio β increases and the problem gets harder). We also observe that all cross-validation methods become somewhat less accurate when the communities become very unbalanced (large t), whereas the model-based methods are not sensitive to t . Comparing different versions of ECV, while the model-based ECV-dev is usually the best, the two model-free versions are similar most of the time. The most drastic differences between them occur with varying the in-out ratio β under the SBM, with the AUC selection underperforming for very low noise problems (β close to 0) and the sum of squared errors underperforms for high noise levels (large β). A possible explanation for that is that the imputation errors are more sensitive to noise, whereas the AUC may have less discriminative power when the network is very “clean” and all predictions are good even if a larger K is chosen (it never underestimates K and most of the errors are choosing $K = 4$ instead of 3).

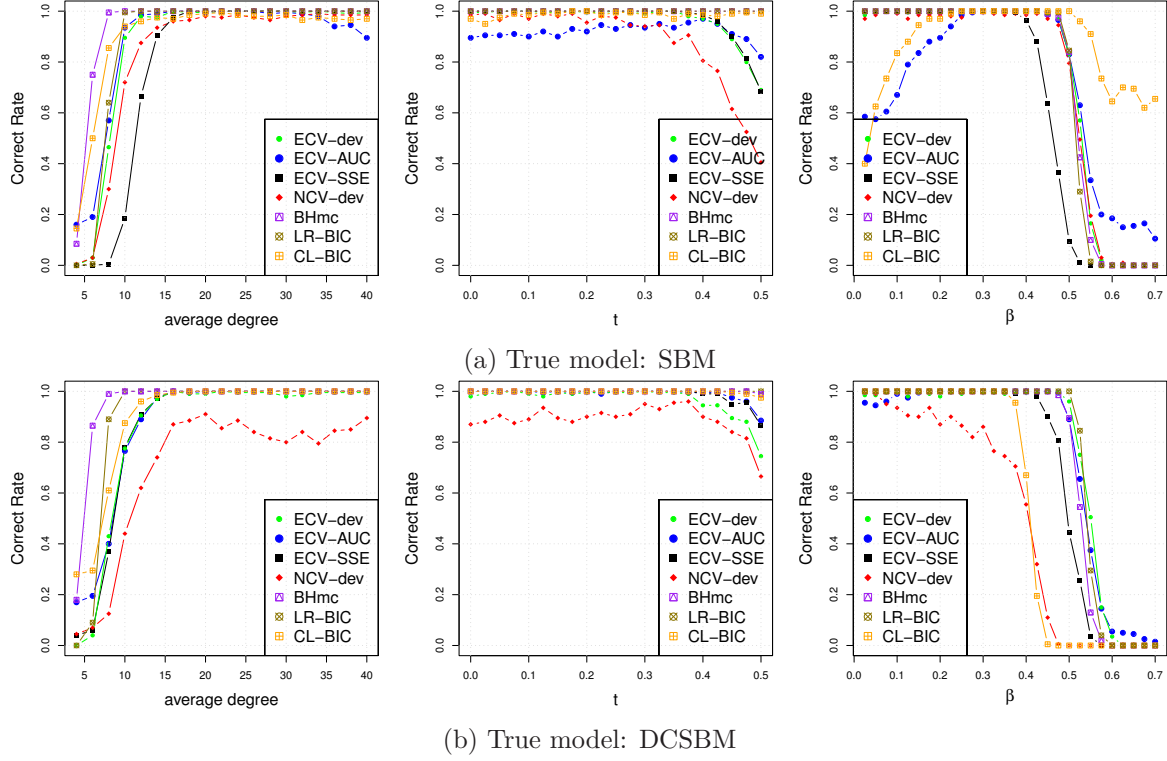


Figure 1: Rate of correct selection of K out of 200 replications under (a) SBM and (b) DCSBM.

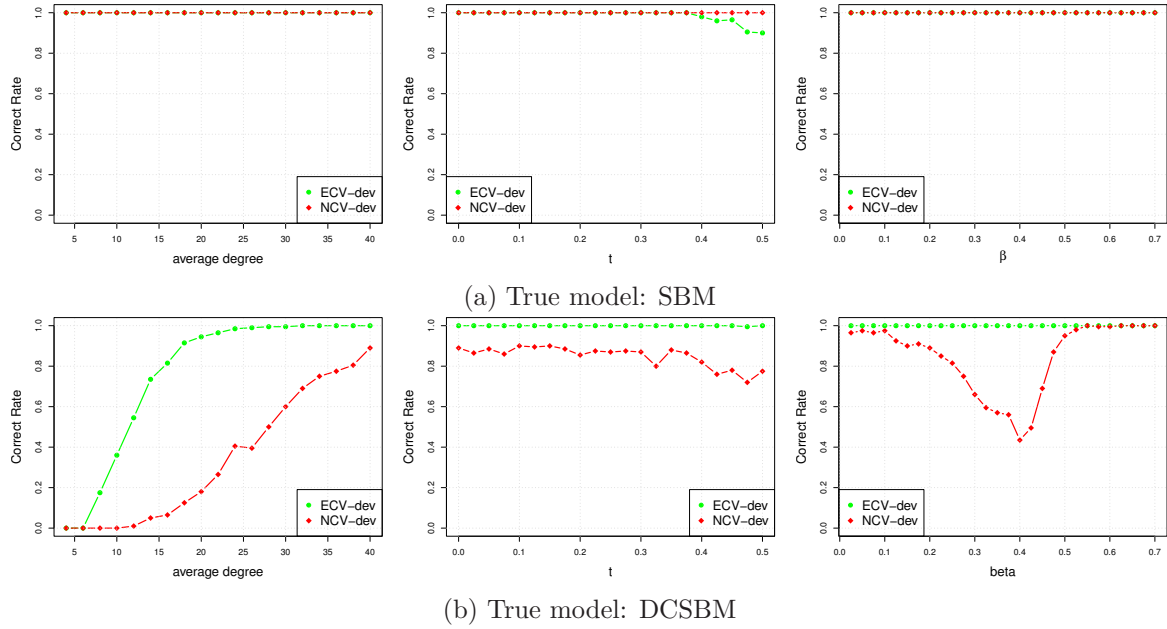


Figure 2: Rate of correct selection between SBM and DCSBM (ignoring K) out of 200 replications under (a) SBM and (b) DCSBM.

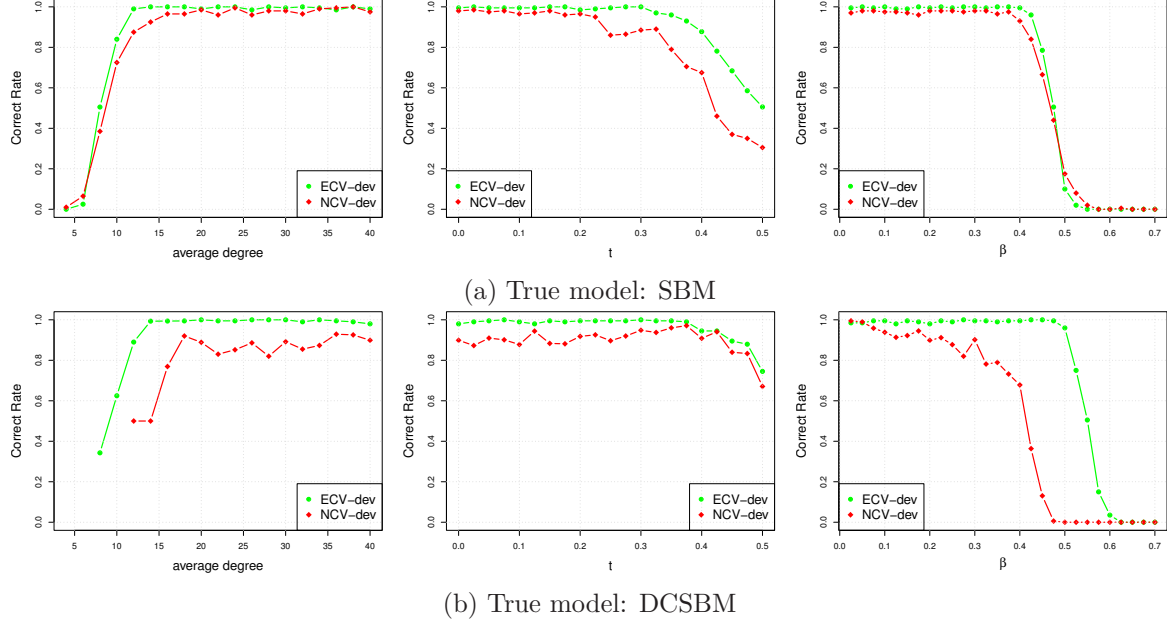


Figure 3: Rate of correct selection of K in cases when the correct model is selected first under (a) SBM and (b) DCSBM.

Figure 2 shows the results of choosing between the SBM and the DCSBM under the SBM (top row) and the DCSBM (bottom row), ignoring which K was chosen for the model afterwards. Only two methods can do this task, NCV-dev and ECV-dev, since the model-free methods cannot compare between different models. Under the SBM, they both perform perfectly under all settings we considered. Under the DCSBM, however, ECV substantially outperforms NCV; for example, for average degree $\lambda = 20$ with $\beta = 0.2, t = 0$, NCV is correct about 20% of the time, and ECV over 90% of the time.

Finally, Figure 3 shows the rate of correctly identifying K only in the cases where the correct model was selected, for the same two methods. We see that again ECV is uniformly better than NCV, and the differences are bigger under the DCSBM again.

4.2 Tuning regularized spectral clustering

Another application of ECV discussed in Section 2.5 is choosing the tuning parameter in regularized spectral clustering. Here we test ECV on networks generated from the DCSBM under the setting described in the previous section, with power law distribution for θ_i , balanced community sizes $\pi = (1/3, 1/3, 1/3)$, out-in ratio $\beta = 0.2$ and average degree $\lambda = 5$, since regularization is generally only relevant when the network is sparse. The candidate set for τ is $\mathcal{C} = \{0.1, 0.2, \dots, 0.9, 1\}$. Without regularization, at this level of sparsity spectral clustering works very poorly (with clustering accuracy of 0.453). Besides ECV, we also compute the average accuracy for each τ in \mathcal{C} (that means, one always uses the same value of τ) as well as DKest tuning of τ proposed by [24].

Figure 4 shows the average accuracy of regularized spectral clustering with these tuning strategies over 200 replications, with error bars indicating ± 2 standard errors. As pointed out in the original paper by [2], as long as τ is not too small, the average accuracy is not very sensitive to τ and the optimal τ value only gives a slightly higher accuracy. However,

tuning τ in a data-driven way may offer protection against a particular bad realization. In this setting, the average accuracy is the highest for ECV, and the DKest method is worse but the difference is small. In our experience, both methods can effectively avoid picking a bad τ under block models, but in real applications, spectral clustering is applied to a wide range of networks many of which do not fit the block models well. In such cases, the model-free ECV is likely to work much better than model-based DKest; two such examples are shown in Section 5.2.

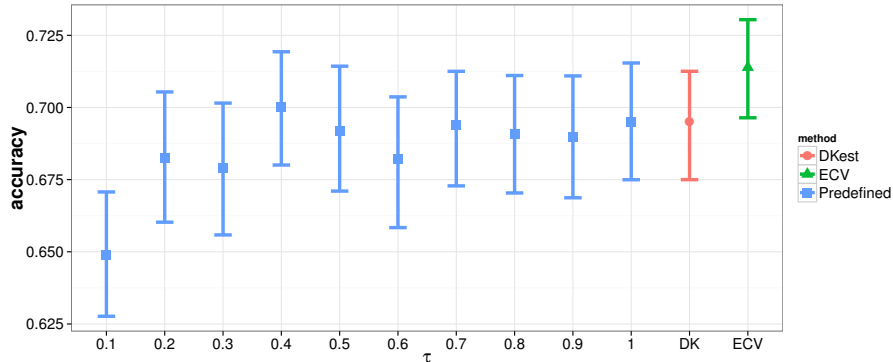


Figure 4: Average clustering accuracy for different fixed values of τ and for DKest and ECV tuning.

4.3 Rank estimation for general networks

The ECV method can also be used to estimate the rank of a general probability matrix from a single network realization, without block model assumptions. To create such a matrix in simulations, we generate an $n \times K$ matrix S with each element drawn independently from the uniform distribution on $(0, 1)$, and set $P = SS^T$. We then normalize P to $[0, 1]$ by dividing by the maximum of P , and generate a network adjacency matrix A with independent Bernoulli edges and $EA = P$. We fix $K = 3$ again and vary the number of nodes n . Table 2 shows results for $n = 700$ and $n = 900$. The methods based on block model likelihood fail in this case, as expected, but the two model-free versions of ECV still work, with ECV-AUC giving perfect results for $n = 900$ and clearly outperforming ECV-SSE in both cases, which also gives a reasonable performance.

Table 2: Frequency of estimated rank values in 200 replications.

n	method	\hat{K} : 1	2	3	4	5	6	7	8
700	ECV-AUC	4	28	168	-	-	-	-	-
	ECV-SSE	28	62	110	-	-	-	-	-
	ECV-dev	1	34	57	20	21	19	15	33
	NCV-dev	-	25	32	22	20	31	27	43
900	ECV-AUC	-	-	200	-	-	-	-	-
	ECV-SSE	-	3	197	-	-	-	-	-
	ECV-dev	-	15	35	32	19	30	28	41
	NCV-dev	-	15	35	17	27	35	33	38

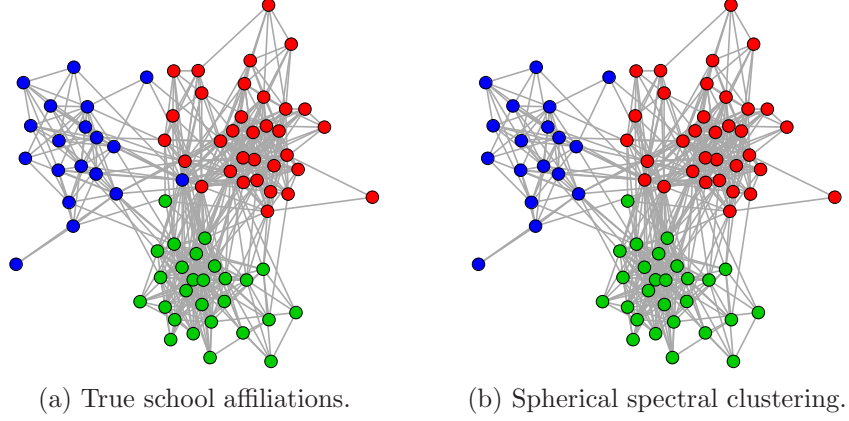


Figure 5: UK faculty friendship network.

5 Applications

In this section, we apply ECV to two real social networks. The first network is a friendship network of faculty at a UK university. The second network is a school friendship network from the National Longitudinal Study of Adolescent Health.

5.1 Community detection on a faculty friendship network

[36] recorded a friendship network of 81 faculty members, available from the package of [14]. The faculty school affiliations are recorded and can be compared to communities detected by various methods. There are four different schools but one of them only contains two people, and thus we deleted these two nodes. The resulting network, shown in Figure 5, has 79 nodes. Node degrees range from 2 to 39, with the average degree of 13.97 and the standard deviation of 7.85.

Since schools can plausibly be expected to correspond to communities, a reasonable question is to select the best block model for this dataset, choosing either SBM or DCSBM and then selecting K . We applied both ECV and NCV to this dataset. As suggested in [11], repeated the validation process 20 times with three folds each time and averaged the results to ensure stability for NCV. Correspondingly, we set $N = 60$ and $p = 0.33^2$ in ECV. The results are shown in Figure 6. In Figure 6 (a), the left vertical axis shows the SSE scale and the right vertical axis the AUC scale. ECV chooses DCSBM with $K = 3$ communities, while NCV picks the SBM with 5 communities. ECV-SSE and ECV-AUC also correctly detect three communities, though they do not address the question of selecting between SBM and DCSBM. DCSBM can be expected to be a better parsimonious model in terms of number of communities for the data as node degrees vary substantially; a similar fit to the data can often be obtained by an SBM with a larger K , as suggested by NCV, but this less parsimonious explanation is usually less interpretable. Choosing the DCSBM also agrees with the higher accuracy of spherical spectral clustering (78/79) compared to spectral clustering (74/79) when clustering into three clusters. Additionally, since the average degree is fairly large, regularization is not helpful for spectral clustering, with ECV selecting $\tau = 0$ and DKest $\tau = 0.2$.

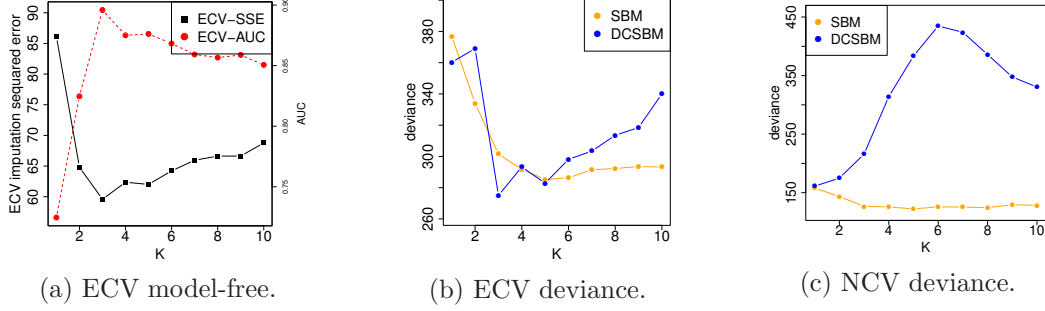


Figure 6: Model selection results of ECV and NCV for the UK faculty data. In (a), the left vertical axis shows the scale of imputation error and the right vertical axis shows the scale of AUC.

5.2 Adolescent school friendship networks

The data come from the National Longitudinal Study of Adolescent Health (the AddHealth study) [18]. AddHealth was a major national longitudinal study of students in grades 7-12 during the school year 1994-1995, after which three further followups were conducted. We use friendship networks constructed from Wave I data¹. Students were asked in a survey to name at most ten friends, up to five of each gender. We consider two students to be friends if one of them named the other, ignoring the direction of edges. Some of the data sets (local communities) included in the study involved two schools, a high school and a middle school that feeds into that high school. The student friendships tend to cluster within their own school, and thus the two schools can be used as ground truth clusters.

We consider two such school networks here, which we call A and B, shown in Figure 7. Network A has 910 nodes and the average node degree is 8.95. Network B has 987 nodes with average node degree of 7.93. The networks are relatively sparse and so spectral clustering may be expected to benefit from regularization. All methods for choosing K we tested earlier in the paper choose $K > 2$ under both SBM and DCSBM. For instance, under DCSBM, ECV estimates $K = 8$ for network A and $K = 9$ for network B; NCV estimates $K = 5$ for network A and $K = 4$ for network B; while LR-BIC estimates $K = 10$ for network A and $K = 7$ for network B. This suggests that there more communities inside each school, as is usually the case in this dataset, and each school’s network is much less homogeneous than a one-block model. However, given that there are two known large communities representing schools in this dataset, we should expect to recover them when applying regularized spectral clustering with $K = 2$. This examples shows that both regularized spectral clustering and cross-validation for tuning are valid even when the block model does not apply.

Table 3 gives the accuracy of regularized spectral clustering for different values of τ ; the values corresponding to the τ picked by ECV and DKest are shown in bold and italic, respectively. Network A needs no regularization to achieve optimal clustering performance. However, if one picks $\tau > 0.5$, the clustering accuracy drops very quickly. This pattern is very different from what we observe in simulation from block models, and it indicates that the model-free nature of the ECV tuning is potentially preferable. In fact, ECV adapts to this situation and picks $\tau = 0.4$ which is only slightly inferior to the optimal value $\tau = 0$, resulting in 880/910 accuracy, while model-based DKest performs poorly, picking $\tau = 1$ which only correctly clusters 635/910 nodes. Network B, on the other hand, does need regularization a little and the optimal value for τ is between 0.3 and 0.6, but the

¹The data can be downloaded from <http://moreno.ss.uci.edu/data.html>

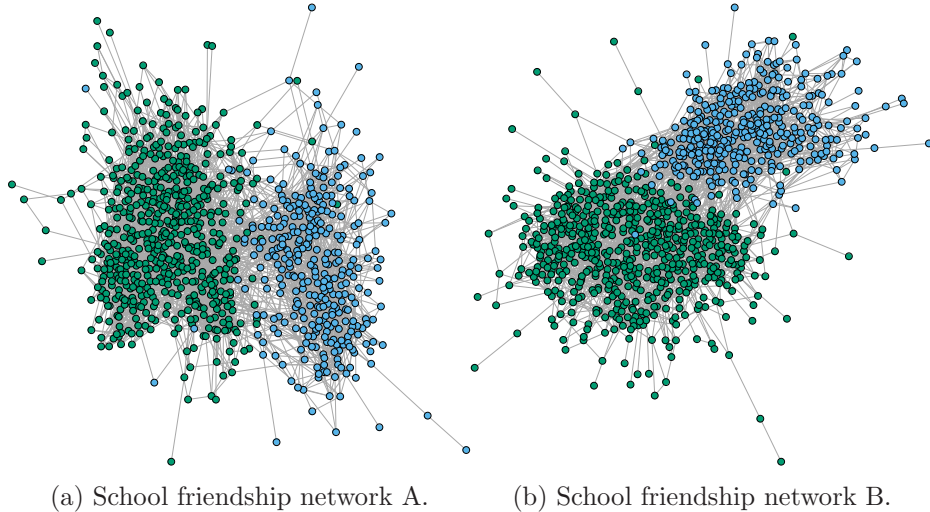


Figure 7: AddHealth networks.

performance is stable for the whole range of $\tau \in [0, 1]$. There is an increase in accuracy for any value of τ compared with no regularization, and the accuracy drops slowly if one keeps increasing τ . ECV selects $\tau = 0.1$ that gives the highest accuracy with 970 nodes out of 983 being correctly clustered, and DKest again picks $\tau = 1$, which misclusters 14 more nodes than $\tau = 0.1$.

Our simulation results and these two real network examples suggest that when the block model is approximately correct and the network is sparse enough to require regularization, the value of τ is not very important as long as it is not too small, and both ECV and DKest perform similarly. However, when the block model does not apply, but there are still clear clusters in the network, regularized spectral clustering is much more sensitive to the choice of τ , and the model-free ECV is more robust than the model-based DKest.

Table 3: Number of correctly clustered nodes for different values of τ in regularized spectral clustering. The values corresponding to the parameters chosen by ECV and DKest are shown in bold and italic, respectively.

τ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
A ($n = 910$)	897	894	891	885	880	871	822	802	725	665	<i>635</i>
B ($n = 987$)	967	970	973	974	974	974	974	969	967	963	<i>956</i>

6 Discussion

We have proposed a general framework for cross-validation on networks based on, in a nutshell, leaving out adjacency matrix entries at random and using matrix completion to fill in the matrix in order to proceed with analysis. While it works well for certain model selection tasks previously studied under the block models, it is applicable and competitive for various other tasks, such as tuning spectral clustering and estimating rank of a general low-rank model. Under the low rank assumption on the underlying probability matrix, we showed that based on the adjacency matrix after randomly removing entries, matrix completion retains the same order of concentration around its mean as the full adjacency matrix; in practice, we expect the method will work well for approximately low rank structures as well.

While our focus was on cross-validation, the general scheme of leaving out entries and matrix completion can be potentially useful for more general scenarios in sampling and bootstrapping networks. Another direction to explore is cross-validation under alternative assumptions to low rank. For example, in the network graphon estimation literature, which are based on the Aldous-Hoover representation of exchangeable networks, [1, 22], some type of graphon smoothness is typically assumed rather than low-rankness [45, 13, 16, 47]. Incorporating these types of assumptions in matrix completion algorithms would allow for cross-validation under this class of models as well.

References

- [1] David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.
- [2] Arash A Amini, Aiyu Chen, Peter J Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.
- [3] Sonia A Bhaskar and Adel Javanmard. 1-bit matrix completion under exact low-rank constraint. In *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, pages 1–6. IEEE, 2015.
- [4] Peter Bickel, David Choi, Xiangyu Chang, Hai Zhang, et al. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943, 2013.
- [5] Peter J Bickel and Purnamrita Sarkar. Hypothesis testing for automated community detection in networks. *arXiv preprint arXiv:1311.2694*, 2013.
- [6] Norbert Binkiewicz, Joshua T Vogelstein, and Karl Rohe. Covariate assisted spectral clustering. *arXiv preprint arXiv:1411.2158*, 2014.
- [7] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.
- [8] Tony Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *The Journal of Machine Learning Research*, 14(1):3619–3647, 2013.
- [9] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- [10] Kamalika Chaudhuri, Fan Chung Graham, and Alexander Tsiatas. Spectral clustering of graphs with general degrees in the extended planted partition model. In *COLT*, volume 23, pages 35–1, 2012.
- [11] Kehui Chen and Jing Lei. Network cross-validation for determining the number of communities in network data. *arXiv preprint arXiv:1411.1715*, 2014.
- [12] Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in the sparse graphs: A spectral algorithm with optimal rate of recovery. *arXiv preprint arXiv:1501.05021*, 2015.
- [13] David Choi and Patrick J Wolfe. Co-clustering separately exchangeable network data. *The Annals of Statistics*, 42(1):29–63, 2014.

- [14] Gabor Csardi. *igraphdata: A Collection of Network Data Sets for the Igraph Package*, 2015. R package version 0.2.2.
- [15] Mark A Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.
- [16] Chao Gao, Yu Lu, and Harrison H Zhou. Rate-optimal graphon estimation. *The Annals of Statistics*, 43(6):2624–2652, 2015.
- [17] Chao Gao, Zongming Ma, Anderson Y Zhang, and Harrison H Zhou. Achieving optimal misclassification proportion in stochastic block model. *arXiv preprint arXiv:1505.03772*, 2015.
- [18] Kathleen Mullan Harris. *The National Longitudinal Study of Adolescent to Adult Health (Add Health), Waves I & II, 1994-1996; Wave III, 2001-2002; Wave IV, 2007-009 [machine-readable data file and documentation]*. Carolina Population Center, University of North Carolina at Chapel Hill, 2009.
- [19] Peter Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems*, pages 657–664, 2008.
- [20] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.
- [21] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic block-models: First steps. *Social networks*, 5(2):109–137, 1983.
- [22] Douglas N Hoover. Relations on probability spaces and arrays of random variables. *Preprint, Institute for Advanced Study, Princeton, NJ*, 2, 1979.
- [23] Jiashun Jin. Fast community detection by SCORE. *The Annals of Statistics*, 43(1):57–89, 2015.
- [24] Antony Joseph and Bin Yu. Impact of regularization on spectral clustering. *The Annals of Statistics*, 44(4):1765–1791, 2016.
- [25] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.
- [26] Raghunandan Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems*, pages 952–960, 2009.
- [27] Olga Klopp. Matrix completion by singular value thresholding: sharp bounds. *Electronic Journal of Statistics*, 9(2):2348–2369, 2015.
- [28] Pierre Latouche, Etienne Birmele, and Christophe Ambroise. Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1):93–115, 2012.
- [29] Can M Le and Elizaveta Levina. Estimating the number of communities in networks by spectral methods. *arXiv preprint arXiv:1507.00827*, 2015.
- [30] Can M Le, Elizaveta Levina, and Roman Vershynin. Concentration and regularization of random graphs. *arXiv preprint arXiv:1506.00669*, 2015.

- [31] Can M Le, Elizaveta Levina, and Roman Vershynin. Sparse random graphs: regularization and concentration of the Laplacian. *arXiv preprint arXiv:1502.03049*, 2015.
- [32] Jing Lei. A goodness-of-fit test for stochastic block models. *The Annals of Statistics*, 44(1):401–424, 2016.
- [33] Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2014.
- [34] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.
- [35] Aaron F McDavid, Thomas Brendan Murphy, Nial Friel, and Neil J Hurley. Improved Bayesian inference for the stochastic block model with application to large networks. *Computational Statistics & Data Analysis*, 60:12–31, 2013.
- [36] Tamás Nepusz, Andrea Petróczy, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107, 2008.
- [37] Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128, 2013.
- [38] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, pages 1878–1915, 2011.
- [39] D. F. Saldana, Y. Yu, and Y. Feng. How many communities are there? *ArXiv e-prints*, December 2014.
- [40] Purnamrita Sarkar and Peter J. Bickel. Role of normalization in spectral clustering for stochastic blockmodels. *Ann. Statist.*, 43(3):962–990, 06 2015.
- [41] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463), 2003.
- [42] Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.
- [43] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
- [44] YX Wang and Peter J Bickel. Likelihood-based model selection for stochastic block models. *arXiv preprint arXiv:1502.02069*, 2015.
- [45] Patrick J Wolfe and Sofia C Olhede. Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*, 2013.
- [46] Xiaoran Yan, Cosma Shalizi, Jacob E Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborova, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(5):P05007, 2014.

- [47] Yuan Zhang, Elizaveta Levina, and Ji Zhu. Estimating network edge probabilities by neighborhood smoothing. *arXiv preprint arXiv:1509.08588*, 2015.
- [48] Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, pages 2266–2292, 2012.

A Proofs

We start with additional notations. For any vector \mathbf{x} , we use $\|\mathbf{x}\|$ to denote its Euclidean norm. We denote the singular values of a matrix M by $\sigma_1(M) \geq \sigma_2(M) \geq \dots \sigma_K(M) > \sigma_{K+1}(M) = \sigma_{K+2}(M) \dots \sigma_n(M) = 0$, where $K = \text{rank}(M)$. Recall the Frobenius norm $\|M\|_F$ is defined by $\|M\|_F^2 = \sum_{ij} M_{ij}^2 = \sum_i \sigma_i(M)^2$, the spectral norm $\|M\| = \sigma_1(M)$, the infinity norm $\|M\|_\infty = \max_{ij} |M_{ij}|$, and the nuclear norm $\|M\|_* = \sum_i \sigma_i(M)$ be the nuclear norm. In addition, the max norm of M [?] is defined as

$$\|M\|_{\max} = \min_{M=UV^T} \max(\|U\|_{2,\infty}^2, \|V\|_{2,\infty}^2),$$

where $\|U\|_{2,\infty} = \max_i (\sum_j U_{ij}^2)^{1/2}$.

We will need the following well-known inequalities:

$$\|M\| \leq \|M\|_F \leq \sqrt{K} \|M\|, \quad (7)$$

$$\|M\|_F \leq \|M\|_* \leq \sqrt{K} \|M\|_F \quad (8)$$

$$|\text{tr}(M_1^T M_2)| \leq \|M_1\| \|M_2\|_* \quad (9)$$

$$\|M\|_{\max} \leq \sqrt{K} \|M\|_\infty. \quad (10)$$

Relationship (9), which holds for any two matrices M_1, M_2 with matching dimensions, is called norm duality for the spectral norm and the nuclear norm [7]. Relationship (10) can be found in [?]. The last one we need is the variational property of spectral norm:

$$\|M\| = \max_{\mathbf{x}, \mathbf{y} \in \mathbb{R}^n: \|\mathbf{x}\| = \|\mathbf{y}\| = 1} \mathbf{y}^T M \mathbf{x}. \quad (11)$$

Our proof will rely on a concentration result for the adjacency matrix. To the best of our knowledge, Lemma 1 stated next is the best concentration bound currently available, proved by [33]. The same concentration was also obtained by [12] and [30].

Lemma 1. Let A be the adjacency matrix of a random graph on n nodes with independent edges. Set $\mathbb{E}(A) = P = [p_{ij}]_{n \times n}$ and assume that $n \max_{ij} p_{ij} \leq d$ for $d \geq C_0 \log n$ and $C_0 > 0$. Then for any $\delta > 0$, there exists a constant $C = C(\delta, C_0)$ such that

$$\|A - P\| \leq C\sqrt{d}$$

with probability at least $1 - n^{-\delta}$.

Another tool we need is the discrepancy between a bounded matrix and its partially observed version given in Lemma 2, which can be viewed as a generalization of Theorem 4.1 of [?] and Lemma 6.4 of [3] to the more realistic uniform missing mechanism in the matrix

completion problem. Let $G \in \mathbb{R}^{n \times n}$ be the indicator matrix associated with the hold-out set Ω , such that if $(i, j) \in \Omega$, $G_{ij} = 0$ and otherwise $G_{ij} = 1$. Note that under the uniform missing mechanism, G can be viewed as an adjacency matrix of an Erdős-Renyi random graph where all edges appear independently with probability p . Note that $P_\Omega A = A \circ G$ where \circ is the Hadamard (element-wise) matrix product.

Lemma 2. Let G an adjacency matrix of an Erdős-Renyi graph with the probability of edge $p \geq C_0 \log n/n$ for a constant C_0 . Then for any $\delta > 0$, with probability at least $1 - n^{-\delta}$, the following relationship holds for any $Z \in \mathbb{R}^{n \times n}$ with $\text{rank}(Z) \leq K$

$$\left\| \frac{1}{p} Z \circ G - Z \right\| \leq C \sqrt{\frac{nK}{p}} \|Z\|_\infty$$

where $C = C(\delta, C_0)$ is a constant that only depends on δ and C_0 .

Proof of Lemma 2. Let $Z = UV^T$, where $U \in \mathbb{R}^{n \times K}$ and $V \in \mathbb{R}^{n \times K}$ are the matrices that achieve the minimum in the definition of $\|Z\|_{\max}$. Denote the ℓ th column of U by U_ℓ and the ℓ th row by U_ℓ^T .

Given any unit vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we have

$$\begin{aligned} \mathbf{y}^T \left(\frac{1}{p} Z \circ G - Z \right) \mathbf{x} &= \sum_\ell \left[\frac{1}{p} \mathbf{y}^T (U_\ell V_\ell^T) \circ G \mathbf{x} - (\mathbf{y}^T U_\ell) (\mathbf{x}^T V_\ell) \right] \\ &= \sum_\ell \left[\frac{1}{p} (\mathbf{y} \circ U_\ell)^T G (\mathbf{x} \circ V_\ell) - (\mathbf{y}^T U_\ell) (\mathbf{x}^T V_\ell) \right]. \end{aligned} \quad (12)$$

Let $\tilde{\mathbf{1}} = \mathbf{1}_n / \sqrt{n}$ be the constant unit vector. For any $1 \leq \ell \leq n$, let $\mathbf{y} \circ U_\ell = \alpha_\ell \tilde{\mathbf{1}} + \beta_\ell \tilde{\mathbf{1}}_\perp^\ell$ in which $\tilde{\mathbf{1}}_\perp^\ell$ is a vector that is orthogonal to $\tilde{\mathbf{1}}$. It is easy to check that

$$\alpha_\ell = (\mathbf{y} \circ U_\ell)^T \tilde{\mathbf{1}} = \frac{1}{\sqrt{n}} \mathbf{y}^T U_\ell.$$

Similarly, we also have

$$(\mathbf{x} \circ V_\ell)^T \tilde{\mathbf{1}} = \frac{1}{\sqrt{n}} \mathbf{x}^T V_\ell.$$

Let $\bar{G} = p \mathbf{1} \mathbf{1}^T$ be the expectation of G with respect to the missing mechanism. Then

$$\begin{aligned} (\mathbf{y} \circ U_\ell)^T G (\mathbf{x} \circ V_\ell) &= \frac{1}{\sqrt{n}} (\mathbf{y}^T U_\ell) \tilde{\mathbf{1}}^T G (\mathbf{x} \circ V_\ell) + \beta_\ell \tilde{\mathbf{1}}_\perp^{\ell T} G (\mathbf{x} \circ V_\ell) \\ &= \frac{1}{\sqrt{n}} (\mathbf{y}^T U_\ell) \tilde{\mathbf{1}}^T \bar{G} (\mathbf{x} \circ V_\ell) + \frac{1}{\sqrt{n}} (\mathbf{y}^T U_\ell) \tilde{\mathbf{1}}^T (G - \bar{G}) (\mathbf{x} \circ V_\ell) + \beta_\ell \tilde{\mathbf{1}}_\perp^{\ell T} G (\mathbf{x} \circ V_\ell). \end{aligned} \quad (13)$$

Notice that $\tilde{\mathbf{1}}^T \bar{G} = np \tilde{\mathbf{1}}^T$, and therefore

$$\frac{1}{\sqrt{n}} (\mathbf{y}^T U_\ell) \tilde{\mathbf{1}}^T \bar{G} (\mathbf{x} \circ V_\ell) = \frac{np}{\sqrt{n}} (\mathbf{y}^T U_\ell) \tilde{\mathbf{1}}^T (\mathbf{x} \circ V_\ell) = p (\mathbf{y}^T U_\ell) (\mathbf{x}^T V_\ell)$$

Further, since $\bar{G}\tilde{\mathbf{1}}_{\perp}^{\ell} = 0$ for any ℓ , we can rewrite (13) as

$$\begin{aligned} (\mathbf{y} \circ U_{\ell})^T G(\mathbf{x} \circ V_{\ell}) &= p(\mathbf{y}^T U_{\ell})(\mathbf{x}^T V_{\ell}) \\ &\quad + \frac{1}{\sqrt{n}}(\mathbf{y}^T U_{\ell})\tilde{\mathbf{1}}^T(G - \bar{G})(\mathbf{x} \circ V_{\ell}) + \beta_{\ell}\tilde{\mathbf{1}}_{\perp}^{\ell T}(G - \bar{G})(\mathbf{x} \circ V_{\ell}). \end{aligned} \quad (14)$$

Substituting (14) into (12) and applying (11) and the Cauchy-Schwarz inequality leads to

$$\begin{aligned} \mathbf{y}^T \left(\frac{1}{p} P_{\Omega} Z - Z \right) \mathbf{x} &= \frac{1}{p} \sum_{\ell} \left[\frac{1}{\sqrt{n}} (\mathbf{y}^T U_{\ell}) \tilde{\mathbf{1}}^T (G - \bar{G})(\mathbf{x} \circ V_{\ell}) + \beta_{\ell} \tilde{\mathbf{1}}_{\perp}^{\ell T} (G - \bar{G})(\mathbf{x} \circ V_{\ell}) \right] \\ &\leq \frac{1}{p} \|G - \bar{G}\| \left[\sum_{\ell} \frac{1}{\sqrt{n}} |\mathbf{y}^T U_{\ell}| \|\mathbf{x} \circ V_{\ell}\| + \sum_{\ell} |\beta_{\ell}| \|\mathbf{x} \circ V_{\ell}\| \right] \\ &\leq \frac{1}{p} \|G - \bar{G}\| \left[\frac{1}{\sqrt{n}} \sqrt{\sum_{\ell} (\mathbf{y}^T U_{\ell})^2} \sqrt{\sum_{\ell} \|\mathbf{x} \circ V_{\ell}\|^2} + \sqrt{\sum_{\ell} \beta_{\ell}^2} \sqrt{\sum_{\ell} \|\mathbf{x} \circ V_{\ell}\|^2} \right]. \end{aligned} \quad (15)$$

Using Cauchy-Schwarz inequality, the definition of max norm and the relationship (10), we get

$$\sum_{\ell} (\mathbf{y}^T U_{\ell})^2 \leq \sum_{\ell} \|\mathbf{y}\|^2 \|U_{\ell}\|^2 = \|U\|_F^2 \leq n \|U\|_{2,\infty}^2 \leq n \|Z\|_{\max} \leq n \sqrt{K} \|Z\|_{\infty}. \quad (16)$$

Similarly,

$$\begin{aligned} \sum_{\ell} \beta_{\ell}^2 &= \sum_{\ell} (\tilde{\mathbf{1}}_{\perp}^{\ell T} (\mathbf{y} \circ U_{\ell}))^2 \leq \sum_{\ell} \|\mathbf{y} \circ U_{\ell}\|^2 \\ &= \sum_{\ell} \sum_i y_i^2 U_{i\ell}^2 \leq \|U\|_{2,\infty}^2 \sum_i y_i^2 \leq \|Z\|_{\max} \leq \sqrt{K} \|Z\|_{\infty}. \end{aligned} \quad (17)$$

We also have

$$\sum_{\ell} \|\mathbf{x} \circ V_{\ell}\|^2 \leq \sqrt{K} \|Z\|_{\infty}. \quad (18)$$

Combining (16), (17) and (18) with (15), we get

$$\mathbf{y}^T \left(\frac{1}{p} P_{\Omega} Z - Z \right) \mathbf{x} \leq \frac{2\sqrt{K}}{p} \|G - \bar{G}\| \|Z\|_{\infty}. \quad (19)$$

From (11), we have

$$\left\| \frac{1}{p} P_{\Omega} Z - Z \right\| = \sup_{\|\mathbf{x}\|=\|\mathbf{y}\|=1} \mathbf{y}^T \left(\frac{1}{p} P_{\Omega} Z - Z \right) \mathbf{x} \leq \frac{2\sqrt{K}}{p} \|G - \bar{G}\| \|Z\|_{\infty}.$$

Finally, Lemma 1 implies

$$\|G - \bar{G}\| \leq C\sqrt{pn} \quad (20)$$

with probability at least $1 - n^{-\delta}$ defined in Lemma 1. Therefore, with probability at least $1 - n^{-\delta}$,

$$\left\| \frac{1}{p} P_{\Omega} Z - Z \right\| \leq C \sqrt{\frac{nK}{p}} \|Z\|_{\infty}.$$

□

With the help of Lemma 2, we can now bound the difference between two feasible points of problem (1).

Lemma 3. Let G be the indicator matrix associated with Ω and the probability of sampling an element $p \geq C_0 \log n/n$ for some constant C_0 . Then with probability at least $1 - n^{-\delta}$, we have the following property for any two feasible points M_1, M_2 of problem (1)

$$\|P_\Omega(M_1 - M_2)\|_F \geq \frac{p}{\sqrt{2K}} \|M_1 - M_2\|_F - 2\sqrt{2}C \sqrt{\frac{Kp}{n}} d \quad (21)$$

where $C = C(\delta, C_0)$ is the same constant as in Lemma 2.

Proof of Lemma 3. Since M_i is a feasible point for (1), $\|M_i\|_\infty \leq \frac{d}{n}, i = 1, 2$ and $\text{rank}(M_1 - M_2) \leq 2K$. According to Lemma 2, with probability at least $1 - n^{-\delta}$, we have

$$\begin{aligned} \left\| \frac{1}{p} P_\Omega(M_1 - M_2) - (M_1 - M_2) \right\| &\leq \left\| \frac{1}{p} P_\Omega(M_1 - M_2) \right\| \\ &\leq C \sqrt{\frac{2nK}{p}} \|M_1 - M_2\|_\infty \leq C \sqrt{\frac{2nK}{p}} (\|M_1\|_\infty + \|M_2\|_\infty) \leq 2\sqrt{2}C \sqrt{\frac{K}{np}} d. \end{aligned}$$

Combining this with (7) further gives

$$\begin{aligned} \frac{\|M_1 - M_2\|_F}{\sqrt{2K}} &\leq \|M_1 - M_2\| \leq \frac{1}{p} \|P_\Omega(M_1 - M_2)\| + 2\sqrt{2}C \sqrt{\frac{K}{np}} d \\ &\leq \frac{1}{p} \|P_\Omega(M_1 - M_2)\|_F + 2\sqrt{2}C \sqrt{\frac{K}{np}} d. \end{aligned}$$

□

Proof of Theorem 1. Recall that we assumed $p \geq C_1 \log n/n$ and $d \geq \max\{d_0, C_2 \log(n)\}$. The inequality in Lemma 2 (and the corresponding bound in Lemma 3) holds with probability at least $1 - n^{-\delta}$ for a constant $C'(\delta, C_1)$. The concentration bound in Lemma 1 holds independently with probability at least $1 - n^{-\delta}$ for a constant $C''(\delta, C_2)$. Thus they both hold with probability at least $1 - 2n^{-\delta}$, and all derivations below are conditional on these inequalities holding.

Let $F(Z) = \|P_\Omega A - P_\Omega Z\|_F^2$. Since the objective function F is quadratic in all entries of Z , we have

$$\begin{aligned} F(Z) &= \sum_{(i,j) \in \Omega} (M_{ij} - A_{ij})^2 + (Z_{ij} - M_{ij})^2 + 2(M_{ij} - A_{ij})(Z_{ij} - M_{ij}) \\ &= \|P_\Omega(Z - M)\|_F^2 + F(M) + 2\text{tr}(P_\Omega(M - A)^T(Z - M)). \end{aligned} \quad (22)$$

From (9),

$$|\text{tr}(P_\Omega(M - A)^T(Z - M))| \leq \|P_\Omega(M - A)\| \cdot \|Z - M\|_*. \quad (23)$$

As both Z and M satisfy the rank constraint, we have $\text{rank}(Z - M) \leq 2K$ and thus by (8),

$$\|Z - M\|_* \leq \sqrt{2K} \|Z - M\|_F. \quad (24)$$

From Lemma 2, we have

$$\frac{1}{p} \|P_\Omega(A - M)\| \leq C'(\delta, C_1) \sqrt{\frac{2nK}{p}} \|A - M\|_\infty + \|A - M\| \leq C'(\delta, C_1) \sqrt{\frac{2nK}{p}} 2\frac{d}{n} + \|A - M\|.$$

Applying the concentration bound in Lemma 1, we further have

$$\begin{aligned} \|P_\Omega(M - A)\| &\leq C' \sqrt{\frac{2nK}{p}} 2\frac{d}{n} p + C'' \sqrt{dp} \leq 2\sqrt{2}C' \sqrt{pK} \sqrt{d} + C'' \sqrt{dp} \\ &\leq \tilde{C}(\delta, C_1, C_2) \sqrt{pK} \cdot \sqrt{d}, \end{aligned} \quad (25)$$

where the second to last inequality is due to $d \leq n$ and $p < 1$ and the last inequality follows from $K \geq 1$. The constant $\tilde{C}(\delta, C_1, C_2) = 2\sqrt{2}C'(\delta, C_1) + C''(\delta, C_2)$.

Combining (24) and (25) leads to

$$|2\text{tr}(P_\Omega(M - A)^T(Z - M))| \leq 2\tilde{C} \sqrt{2pK^2d} \|Z - M\|_F. \quad (26)$$

Combining (22) and (26) gives

$$F(Z) \geq F(M) - 2\tilde{C} \sqrt{2pK^2d} \|Z - M\|_F + \|P_\Omega(Z - M)\|_F^2.$$

In particular, taking $Z = \hat{M}$ leads to

$$0 \geq F(\hat{M}) - F(M) \geq -2\tilde{C} \sqrt{2pK^2d} \|\hat{M} - M\|_F + \|P_\Omega(\hat{M} - M)\|_F^2. \quad (27)$$

To apply Lemma 3, we have to check two cases:

Case 1: $\frac{p\|\hat{M} - M\|_F}{\sqrt{2K}} < 4\sqrt{2}C' \sqrt{\frac{Kp}{n}}d$. This directly gives the bound

$$\|\hat{M} - M\|_F < 8C' \frac{Kd}{\sqrt{np}}. \quad (28)$$

Case 2: $\frac{p\|\hat{M} - M\|_F}{\sqrt{2K}} \geq 4\sqrt{2}C' \sqrt{\frac{Kp}{n}}d$. Then

$$\frac{p\|\hat{M} - M\|_F}{\sqrt{2K}} - 2\sqrt{2}C' \sqrt{\frac{Kp}{n}}d \geq \frac{p\|\hat{M} - M\|_F}{2\sqrt{2K}}. \quad (29)$$

Applying Lemma 3 and (29) to (27) gives

$$\|\hat{M} - M\|_F \left(p^2 \frac{\|\hat{M} - M\|_F}{8K} - 2\tilde{C} \sqrt{2pK^2d} \right) \leq 0$$

which leads to

$$\|\hat{M} - M\|_F \leq 16\sqrt{2}\tilde{C}K^2\sqrt{d}/p^{3/2}. \quad (30)$$

Now combining the two cases, we get

$$\|Z - M\|_F \leq \max \left(16\sqrt{2}\tilde{C}K^2\sqrt{d}/p^{3/2}, 2\sqrt{2}C'Kd/\sqrt{np} \right). \quad (31)$$

Finally, using $d \leq n$, $K \geq 1$ and $p < 1$ we know that the first term dominates the second one in the order of magnitude. This completes the proof. \square

To prove Corollaries 1 and 2, we need the following lemma.

Lemma 4. Under the assumptions of either Corollary 1 or Corollary 2, we have

$$\bar{d}/d_0 = O_P(1)$$

where

$$\bar{d} = n \sum_{(i,j) \in \Omega} A_{ij}/|\Omega|, \quad d_0 = n\|M\|_\infty = n\rho_n\|B_0\|_\infty.$$

Proof of Lemma 4. We prove this under the assumptions of Corollary 1; the proof under the assumptions of Corollary 2 is similar. Since Ω is random and independent of the network, it is enough to show that

$$\frac{\bar{d}}{n} = \frac{\sum_{i < j} A_{ij}}{n(n-1)/2}$$

is of the same order as d_0/n . Define

$$O_{kk'} = \sum_{c_i=k, c_j=k'} A_{ij}$$

and

$$n_{kk'} = |\{i : c_i = k\} \cap \{j : c_j = k'\}|.$$

Note that $O_{kk'} \sim \text{Binomial}(n_{kk'}, B_{kk'})$. Thus by Bernstein's inequality and the definition of ρ_n

$$O_{kk'}/n_{kk'} = B_{kk'} + O_P \left(\sqrt{\frac{\rho_n}{n_{kk'}}} \right) = \rho_n B_{0,kk'} + O_P \left(\sqrt{\frac{\rho_n}{n_{kk'}}} \right).$$

Therefore

$$\begin{aligned} \frac{\bar{d}}{n} &= \frac{\sum_{i < j} A_{ij}}{n(n-1)/2} = \frac{\sum_{k \leq k'} n_{kk'} \cdot O_{kk'}/n_{kk'}}{n(n-1)/2} \\ &= \frac{\sum_{k \leq k'} n_{kk'} \rho_n B_{0,kk'}}{n(n-1)/2} + O_P \left(\frac{\sum_{k \leq k'} \sqrt{n_{kk'} \rho_n}}{n(n-1)/2} \right). \end{aligned} \quad (32)$$

Choosing k_1, k_2 such that $B_{0mk_1k_2} = \|B_0\|_\infty$, we have $n_{k_1k_2} \geq \gamma^2 n^2/2$ according to A4. Thus

$$\frac{\sum_{k \leq k'} n_{kk'} \rho_n B_{0,kk'}}{n(n-1)/2} \geq \frac{\gamma^2 n^2 \rho_n B_{0,k_1k_2}/2}{n(n-1)/2} = \gamma^2 \rho_n B_{0,k_1k_2} \left(1 + O\left(\frac{1}{n}\right) \right). \quad (33)$$

Recall that $\rho_n \geq c \frac{\log(n)}{n}$. Combining (32) and (33) leads to the desired result. In particular, it is clear that the κ in Corollary 1 can be taken to be $(1 + \epsilon) \frac{1}{\gamma^2}$ for a constant ϵ . \square

Proof of Corollaries 1 and 2. Part 1 of Corollary 1 is a direct consequence of Lemma 4 and Theorem 1. Part 2 of Corollary 1 can be proved following the strategy of Corollary 3.2 of [33].

Part 1 of Corollary 2 also follows directly from Lemma 4 and Theorem 1. To prove Part 2 of Corollary 2, recall that $n_k = |\{i : c_i = k\}|$. Following [33], define $\boldsymbol{\theta}_k = \{\theta_i\}_{c_i=k}$ and

$$\nu_k = \frac{1}{n_k^2} \sum_{i:c_i=k} \frac{\|\boldsymbol{\theta}_k\|^2}{\theta_i^2}.$$

Let $\tilde{n}_k = \|\boldsymbol{\theta}_k\|^2$ be the “effective size” of the k th community. Under A5, we have

$$\nu_k \leq \frac{1}{n_k^2} \sum_{i:c_i=k} \frac{n_k}{\theta_0^2} = \frac{1}{\theta_0^2}.$$

Furthermore, when A4 and A5 hold, we have

$$\frac{\sum_k n_k^2 \nu_k^2}{\min_k \tilde{n}_k^2} \leq \frac{\sum_k n_k^2 \nu_k^2}{\min_k n_k^2 \theta_0^4} \leq \frac{\sum_k n_k^2}{\gamma^2 \theta_0^8} \leq \frac{K}{\gamma^2 \theta_0^8} = O(1). \quad (34)$$

Part 2 of Corollary 2 can then be proved by following the proof of Corollary 4.3 of [33] and applying (34). □

B An algorithm for the matrix completion step

As discussed in the paper, the problem (1) has natural convex relaxations which can be solved very efficiently. [34] proposed a hardImpute algorithm to approximately solve the original missing value SVD problem which can be used for large matrices. They reported that when the rank of the underlying true matrix is small, the hardImpute algorithm typically gives better predictions than the softImpute algorithm for nuclear norm relaxation. We apply a similar fixed point iteration method to our problem, and even though we use the primal form instead of the Lagrangian formulation in [34], we still refer to this procedure as hardImpute.

Let P_{Ω^\perp} be the projection onto Ω^\perp so that any matrix A can be written as $A = P_\Omega A + P_{\Omega^\perp} A$. Let $S_H(A, K)$ is the rank- K SVD approximation of A . That is, if the SVD of A is $A = UDV^T$ where $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$, then

$$S_H(A, K) = UD_K V^T,$$

where $D_K = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_K, 0, \dots, 0)$.

The computation is summarized in Algorithm 3.

Algorithm 3 (Constrained hardImpute algorithm). Given A , K , and convergence tolerance ϵ :

1. Initialize with a feasible $M^{(0)}$ and $k = 0$.
2. $M^{(k)} := M^{(k-1)}$.

3. $A^{(k)} := P_{\Omega}A + P_{\Omega^{\perp}}M^{(k-1)}$ and $M^{(k)} := S_H(A^{(k)}, K)$.
4. For all (i, j) , $1 \leq i, j \leq n$, $M_{ij}^{(k)} := \min(1, \max(0, M_{ij}^{(k)}))$ (threshold to between 0 and 1).
5. If $\|M^{(k)} - M^{(k-1)}\|_F < \epsilon$, go to 6; else $k := k + 1$ and return to 2.
6. Return $\hat{M} = M^k$.

We believe any reasonable matrix completion method should work for ECV. The choice of method in practice may depend on computational constraints. For instance, Algorithm 3 requires rank K SVD in each iteration, which may be costly if N is very large, say more than 10^4 . In this case, a faster alternative can be the universal singular value thresholding (USVT) method of [?] that only requires performing the SVD once and also has theoretical guarantees. For smaller N , the performance of USVT is typically inferior. The choice of matrix completion algorithm may also depend on interpretations. In our example of block model selection, the rank gives the number of communities. So in the setting, using the exact rank constraint is more interpretable.